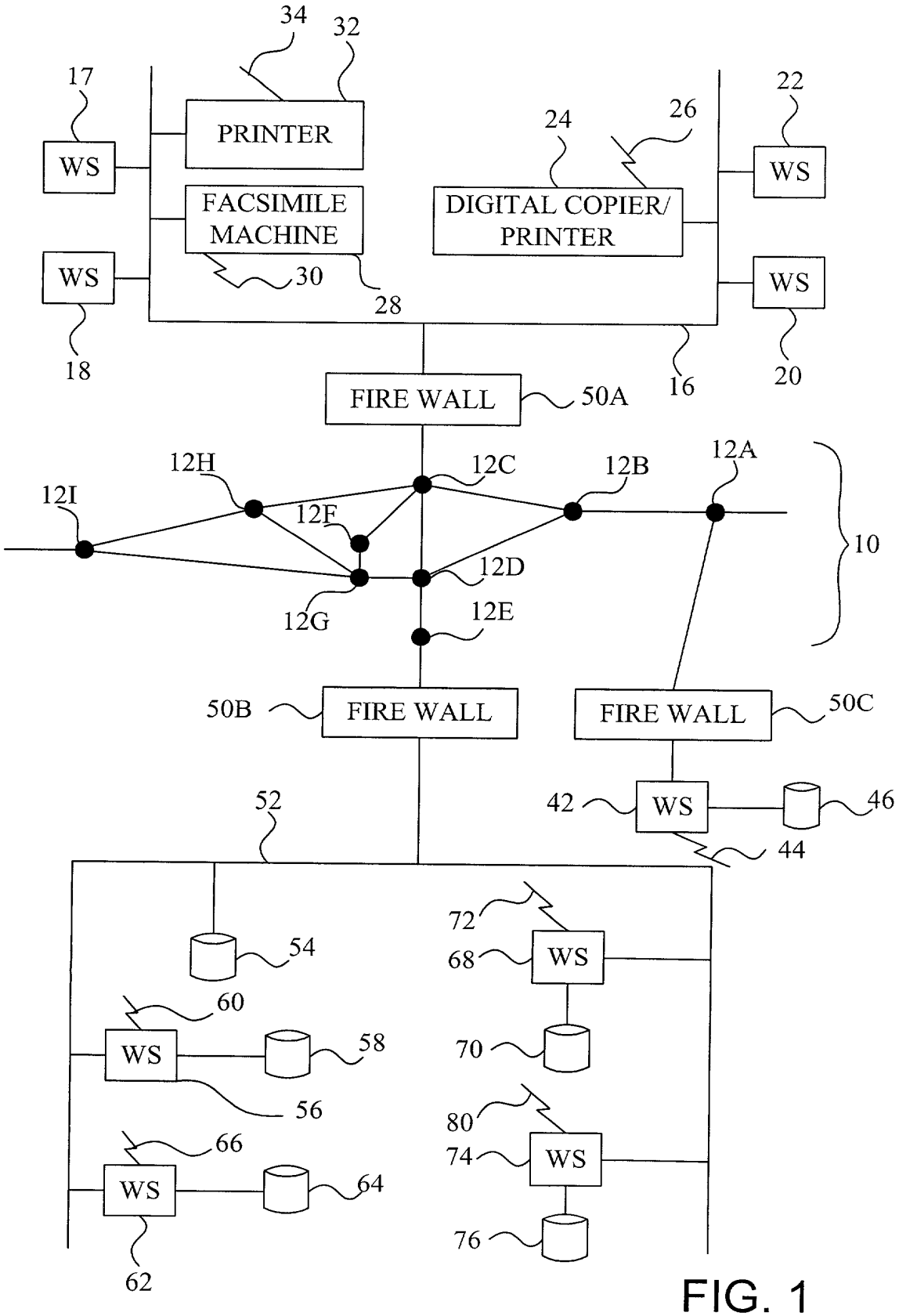

34
32
17
PRINTER
WS
22
24
26
WS
FACSIMILE MACHINE
DIGITAL COPIER/ PRINTER
WS
30
28
18
16
20
FIRE WALL
50A
12H
12C
12B
12A
12I
12F
10
12D
12G
12E
50B
FIRE WALL
FIRE WALL
50C
52
42
WS
46
44
54
72
68
WS
60
WS
58
70
56
80
66
74
WS
WS
64
76
62

FIG. 1

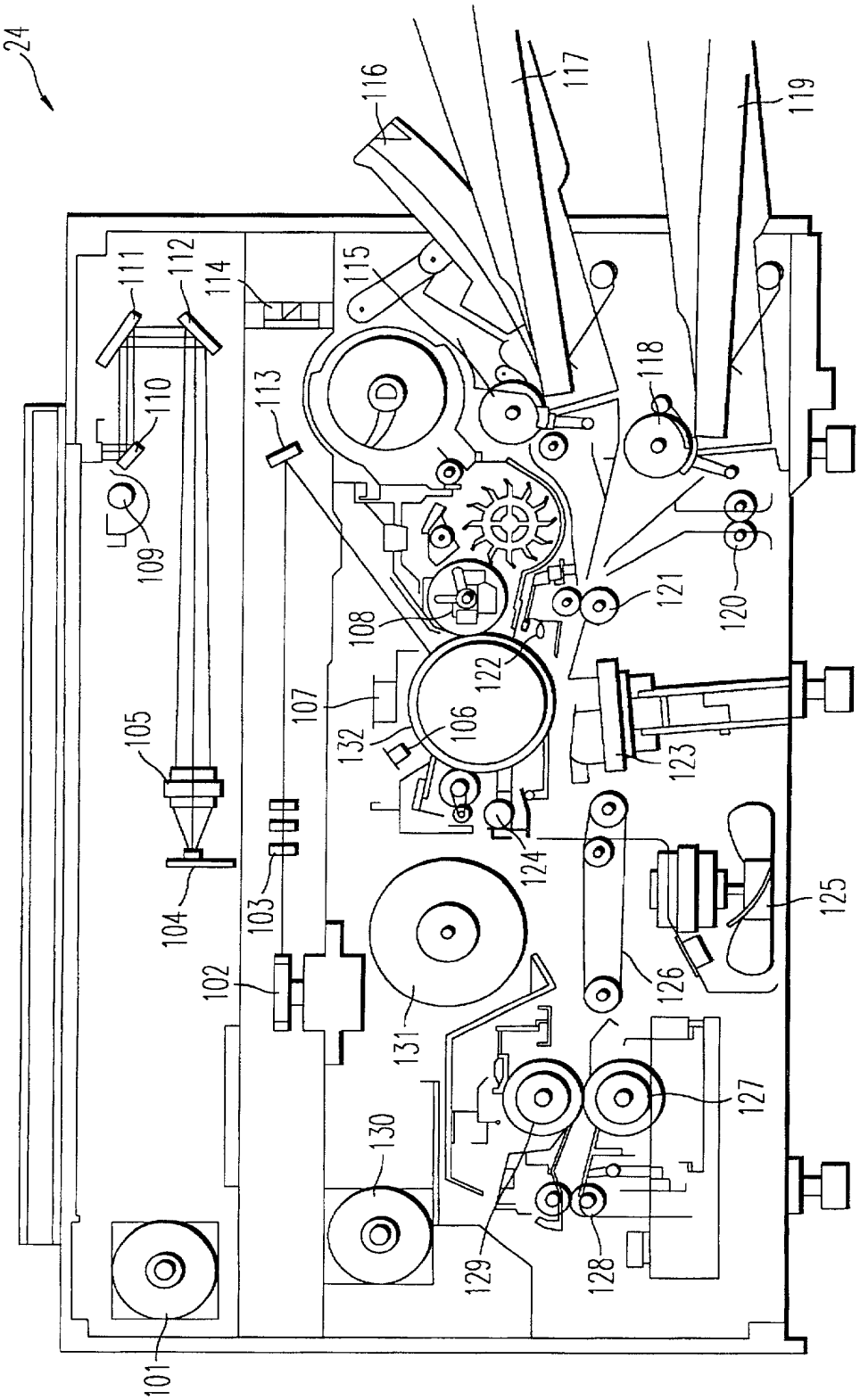

24
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132

FIG. 2

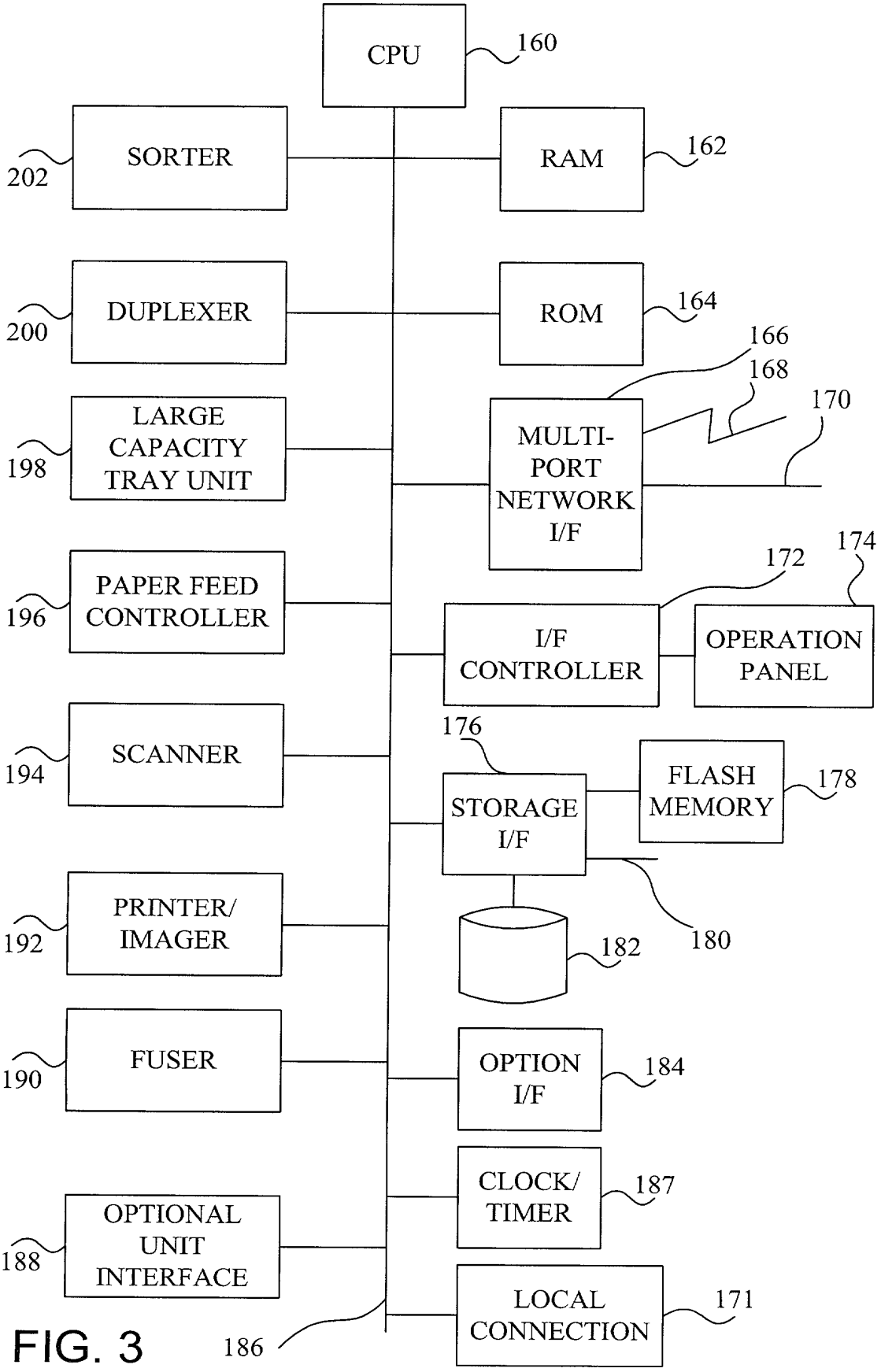

CPU
160
SORTER
202
RAM
162
DUPLEXER
200
ROM
164
166
168
170
LARGE CAPACITY TRAY UNIT
198
MULTI-PORT NETWORK I/F
PAPER FEED CONTROLLER
196
172
174
I/F CONTROLLER
OPERATION PANEL
SCANNER
194
176
FLASH MEMORY
178
STORAGE I/F
PRINTER/ IMAGER
192
182
180
FUSER
190
OPTION I/F
184
CLOCK/ TIMER
187
OPTIONAL UNIT INTERFACE
188
LOCAL CONNECTION
171
186

FIG. 3

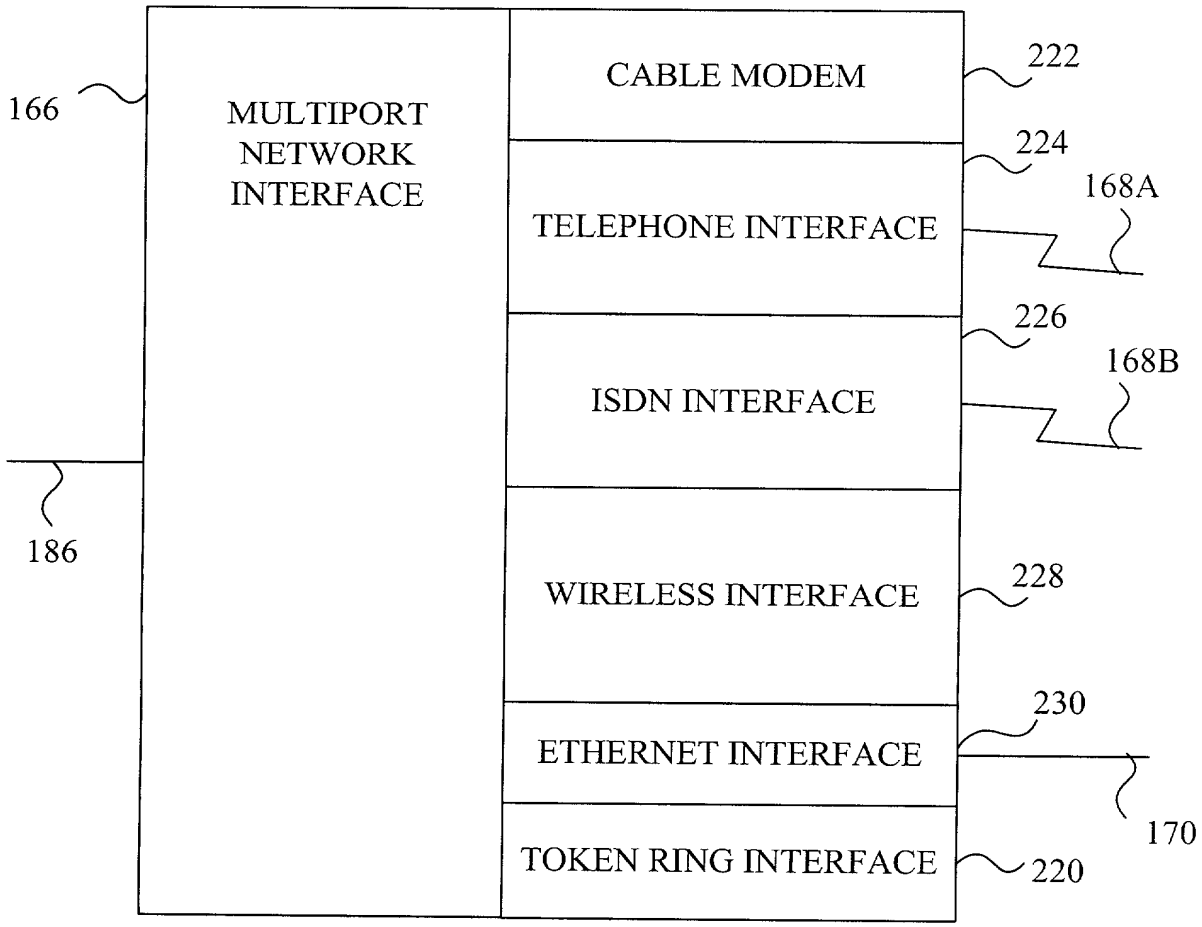

166
MULTIPORT NETWORK INTERFACE
186
CABLE MODEM
222
224
TELEPHONE INTERFACE
168A
226
ISDN INTERFACE
168B
WIRELESS INTERFACE
228
230
ETHERNET INTERFACE
170
TOKEN RING INTERFACE
220

FIG. 4

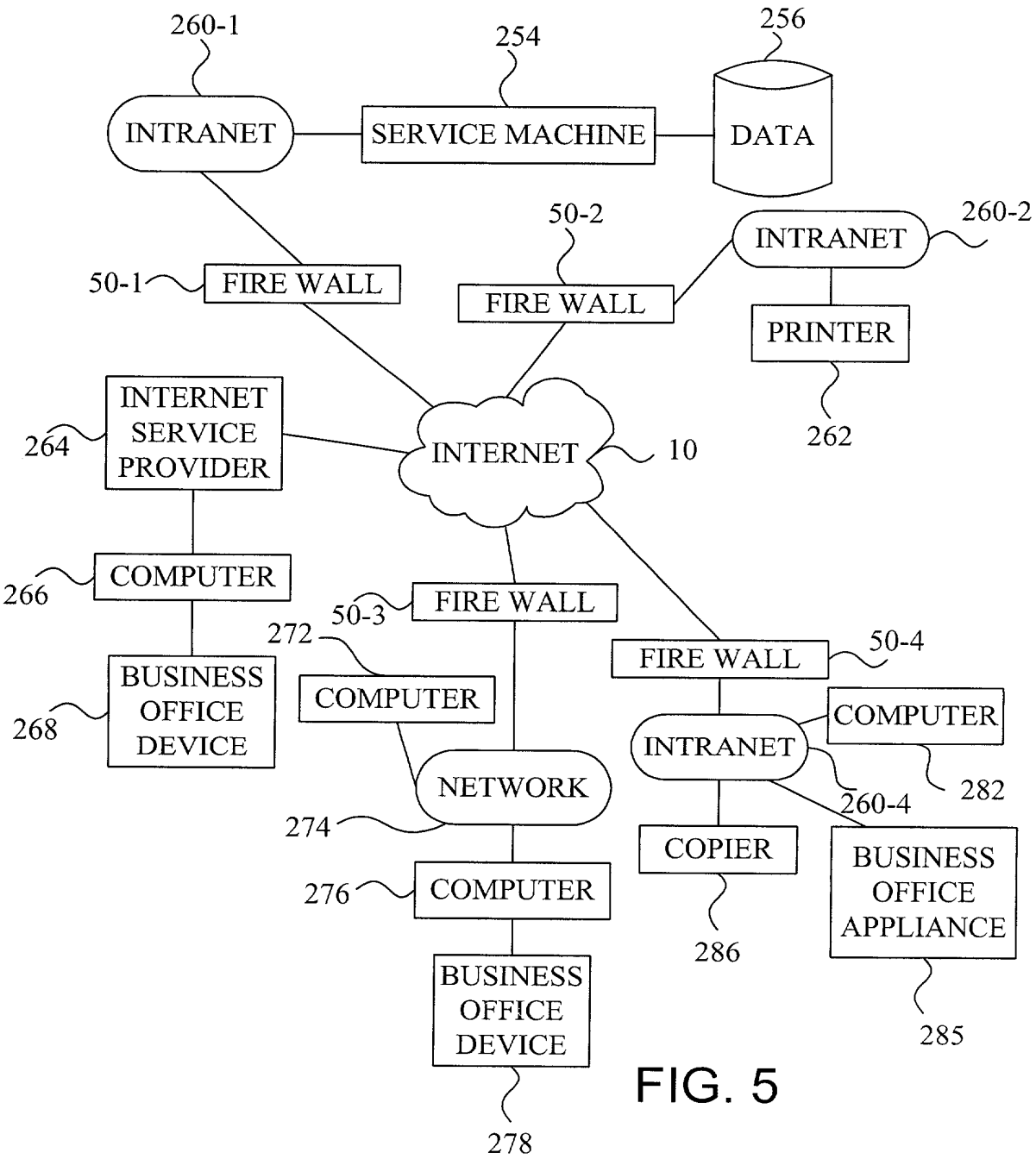

260-1
254
256
INTRANET
SERVICE MACHINE
DATA
50-2
260-2
INTRANET
50-1
FIRE WALL
FIRE WALL
PRINTER
262
INTERNET SERVICE PROVIDER
264
INTERNET
10
COMPUTER
266
50-3
FIRE WALL
272
50-4
FIRE WALL
BUSINESS OFFICE DEVICE
268
COMPUTER
COMPUTER
INTRANET
282
NETWORK
260-4
274
COPIER
BUSINESS OFFICE APPLIANCE
276
COMPUTER
286
BUSINESS OFFICE DEVICE
285
278

FIG. 5

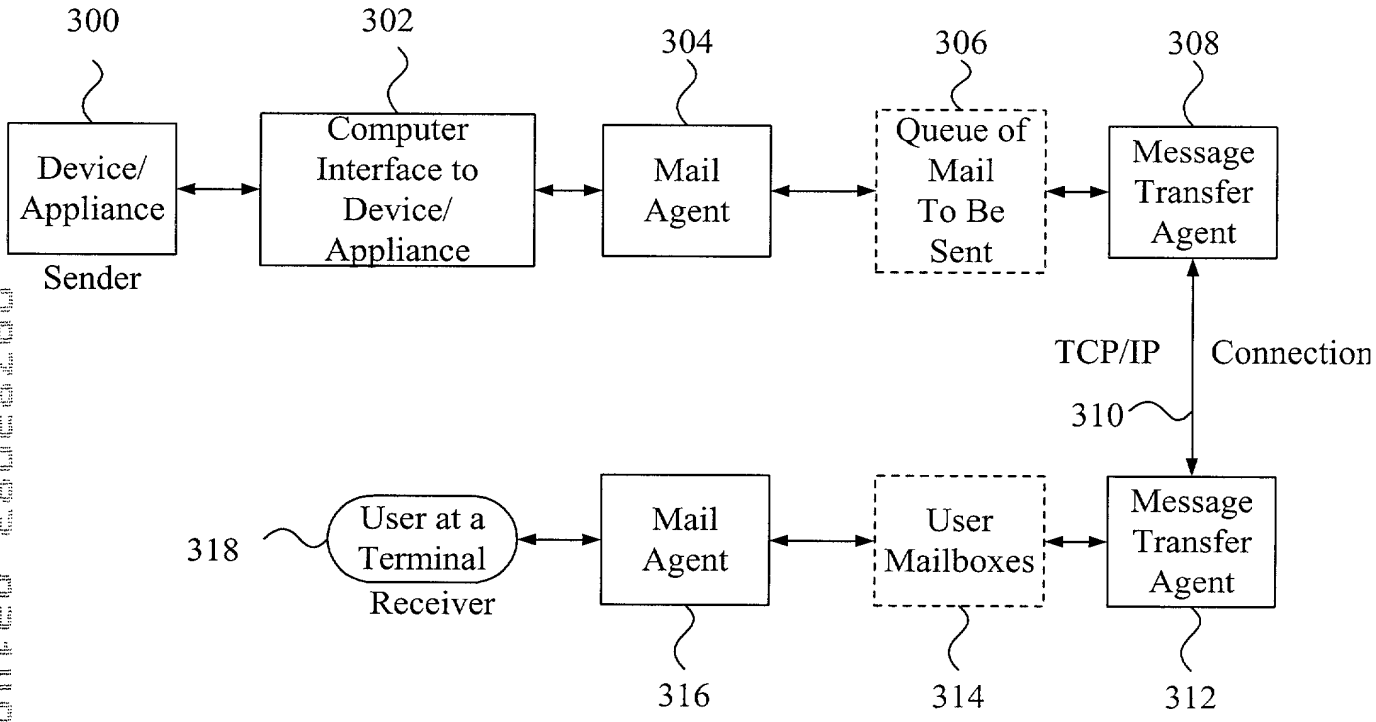

300
302
304
306
308
Device/ Appliance
Sender
Computer Interface to Device/ Appliance
Mail Agent
Queue of Mail To Be Sent
Message Transfer Agent
TCP/IP
Connection
310
318
User at a Terminal
Receiver
Mail Agent
User Mailboxes
Message Transfer Agent
316
314
312

FIG. 6A

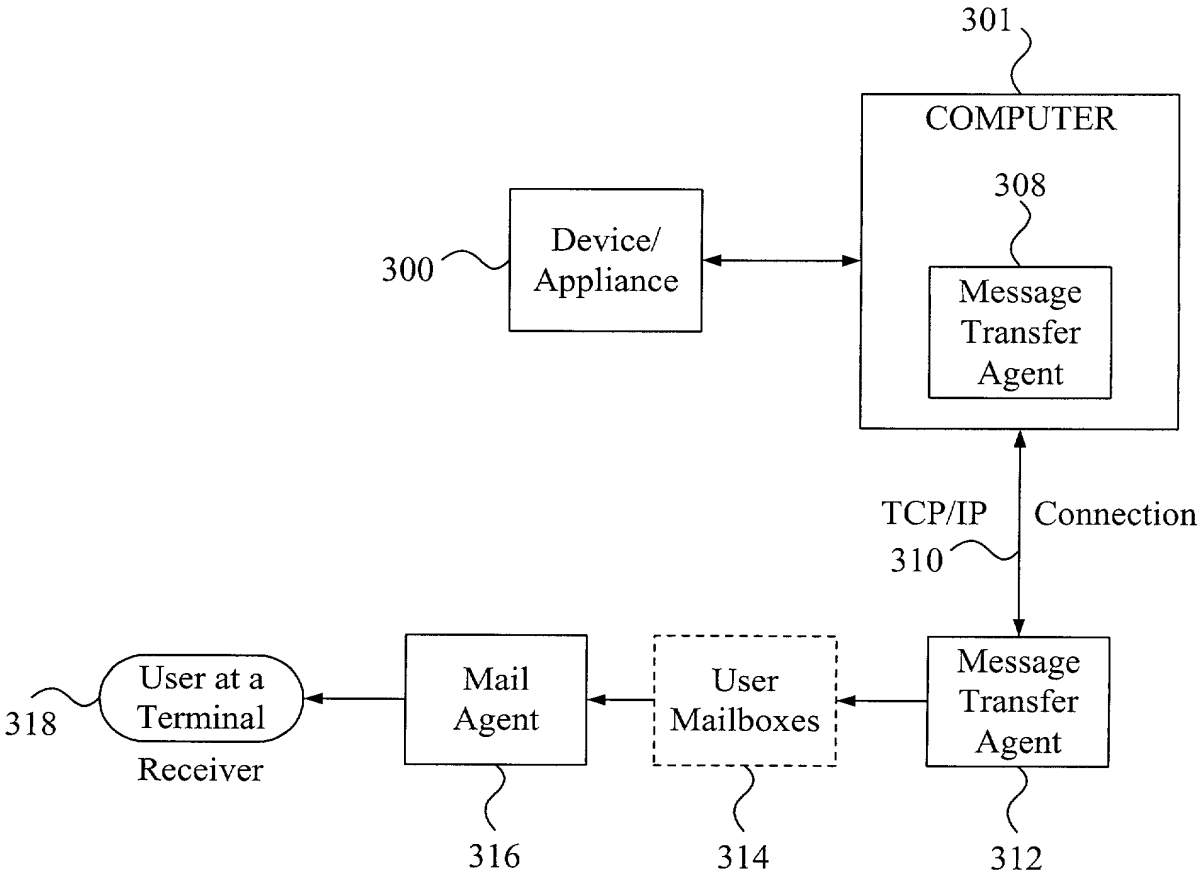

301
COMPUTER
308
Message Transfer Agent
300
Device/ Appliance
TCP/IP Connection
310
Message Transfer Agent
User Mailboxes
Mail Agent
User at a Terminal
318
Receiver
316
314
312

FIG. 6B

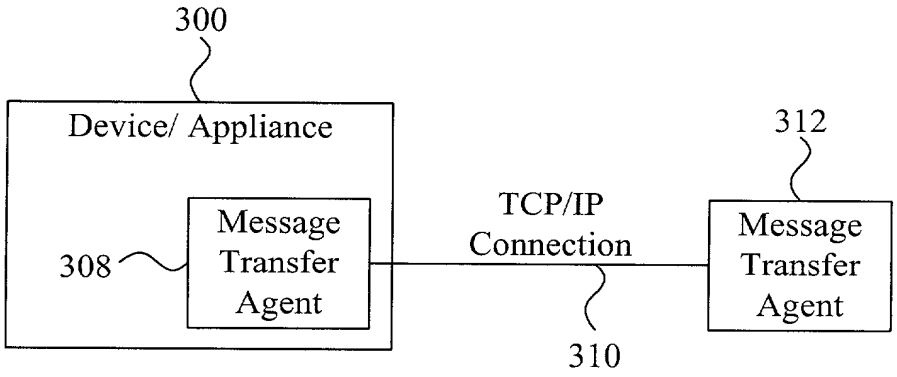

300
Device/ Appliance
312
308
Message Transfer Agent
TCP/IP Connection
Message Transfer Agent
310

FIG. 6C

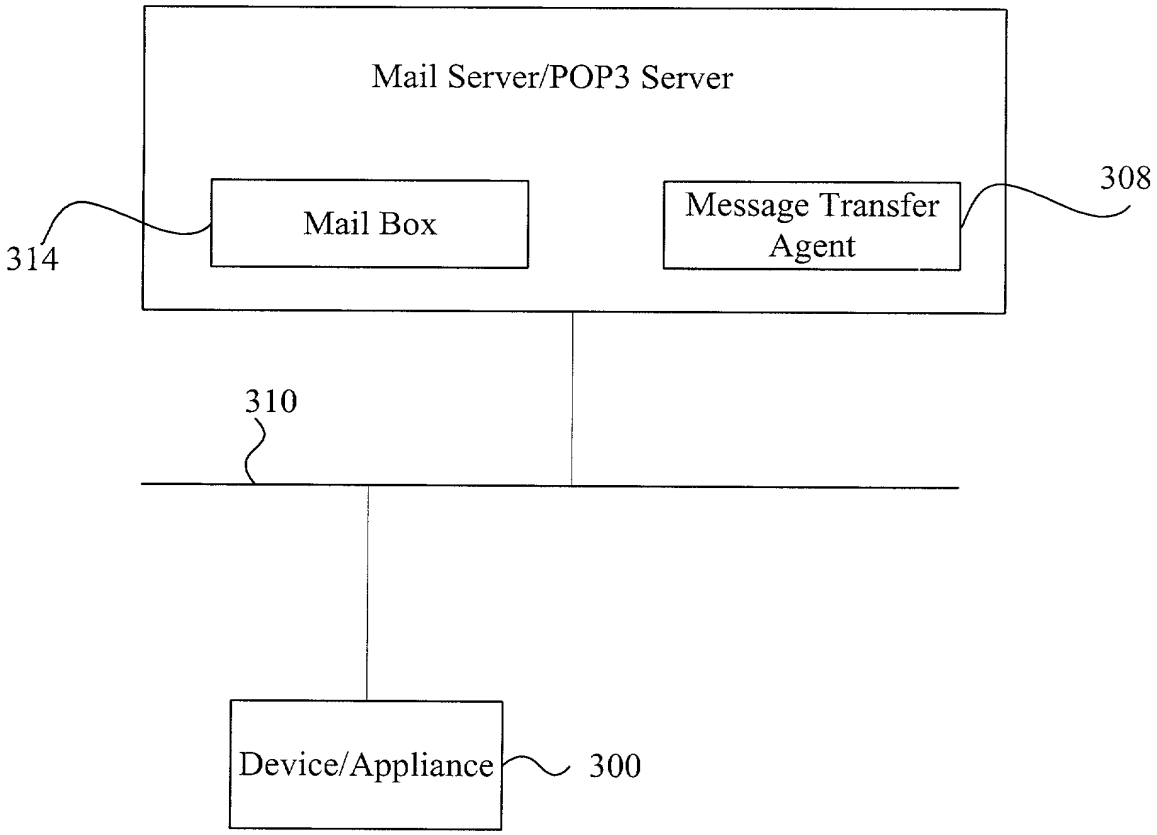

Mail Server/POP3 Server
Mail Box
Message Transfer Agent
308
314
310
Device/Appliance
300

Fig. 6D

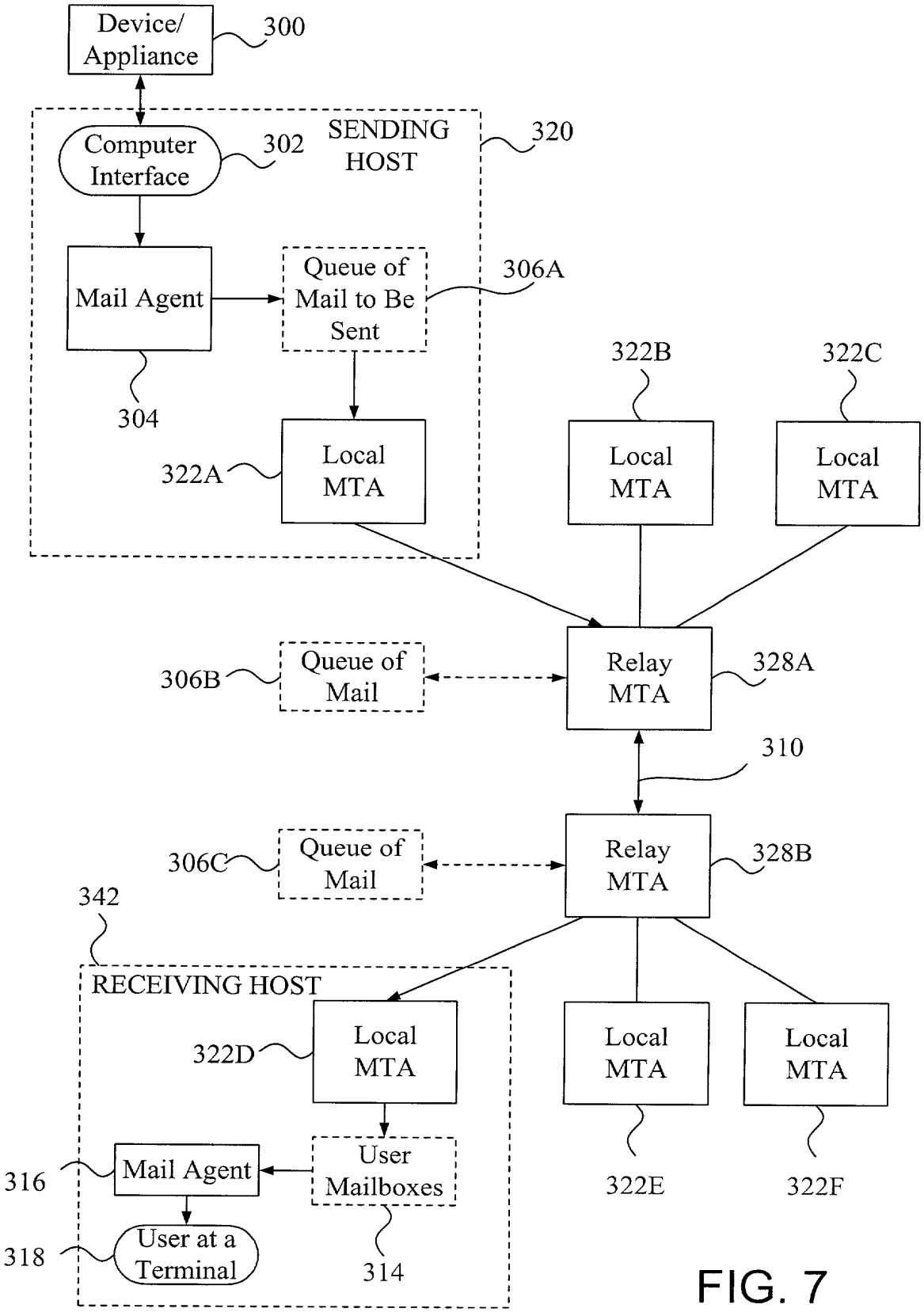

Device/ Appliance
300
SENDING HOST
320
Computer Interface
302
Mail Agent
Queue of Mail to Be Sent
306A
304
322A
Local MTA
322B
Local MTA
322C
Local MTA
306B
Queue of Mail
Relay MTA
328A
310
306C
Queue of Mail
Relay MTA
328B
342
RECEIVING HOST
322D
Local MTA
Local MTA
Local MTA
316
Mail Agent
User Mailboxes
318
User at a Terminal
314
322E
322F

FIG. 7

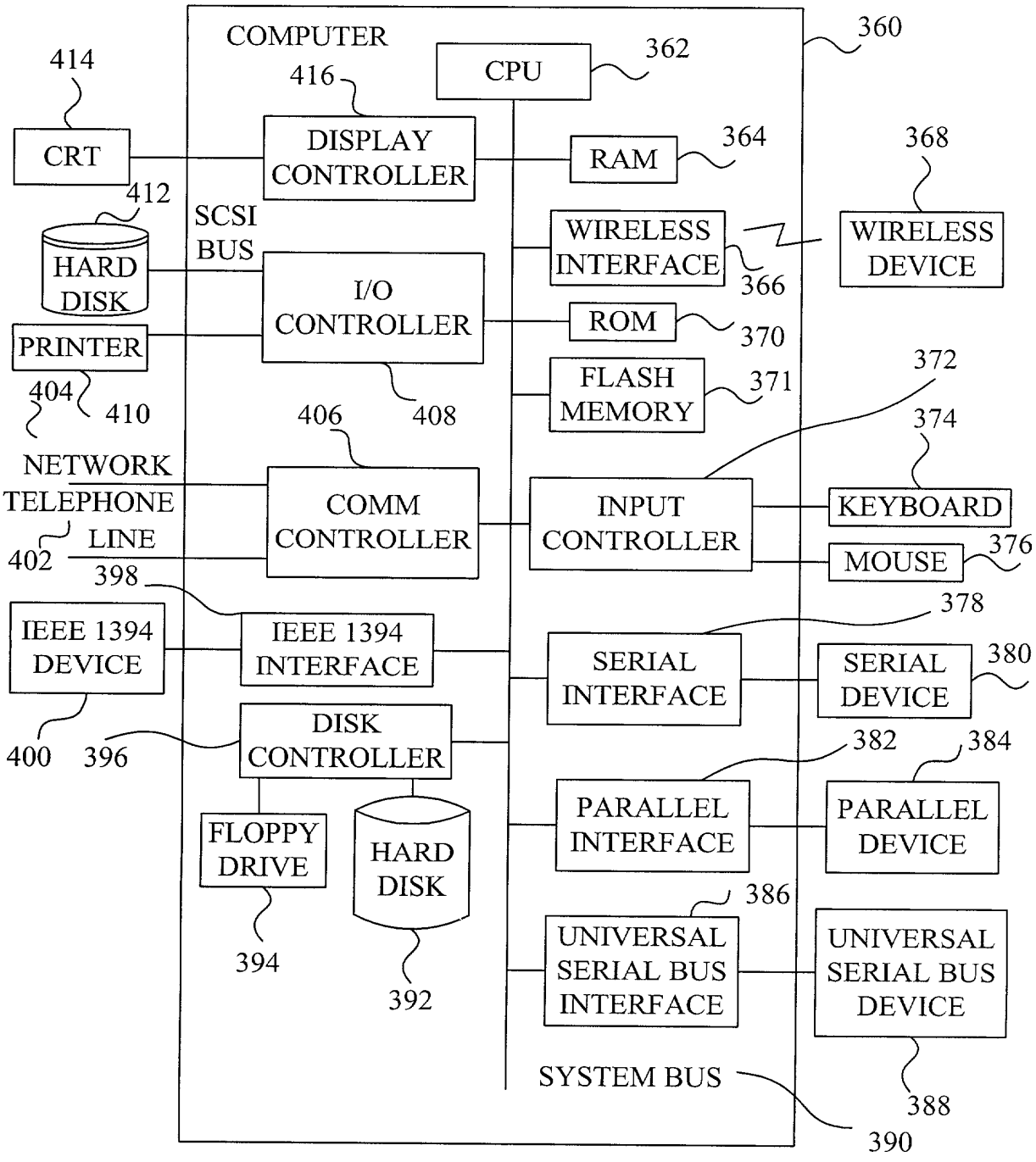

COMPUTER
360
CPU
362
414
416
CRT
DISPLAY CONTROLLER
RAM
364
368
412
SCSI BUS
WIRELESS INTERFACE
WIRELESS DEVICE
HARD DISK
I/O CONTROLLER
366
ROM
370
PRINTER
FLASH MEMORY
371
372
404
410
406
408
NETWORK TELEPHONE LINE
374
COMM CONTROLLER
INPUT CONTROLLER
KEYBOARD
402
MOUSE
376
398
378
IEEE 1394 DEVICE
IEEE 1394 INTERFACE
SERIAL INTERFACE
SERIAL DEVICE
380
400
396
DISK CONTROLLER
382
384
PARALLEL INTERFACE
PARALLEL DEVICE
FLOPPY DRIVE
HARD DISK
386
UNIVERSAL SERIAL BUS INTERFACE
UNIVERSAL SERIAL BUS DEVICE
394
392
SYSTEM BUS
388
390

FIG. 8

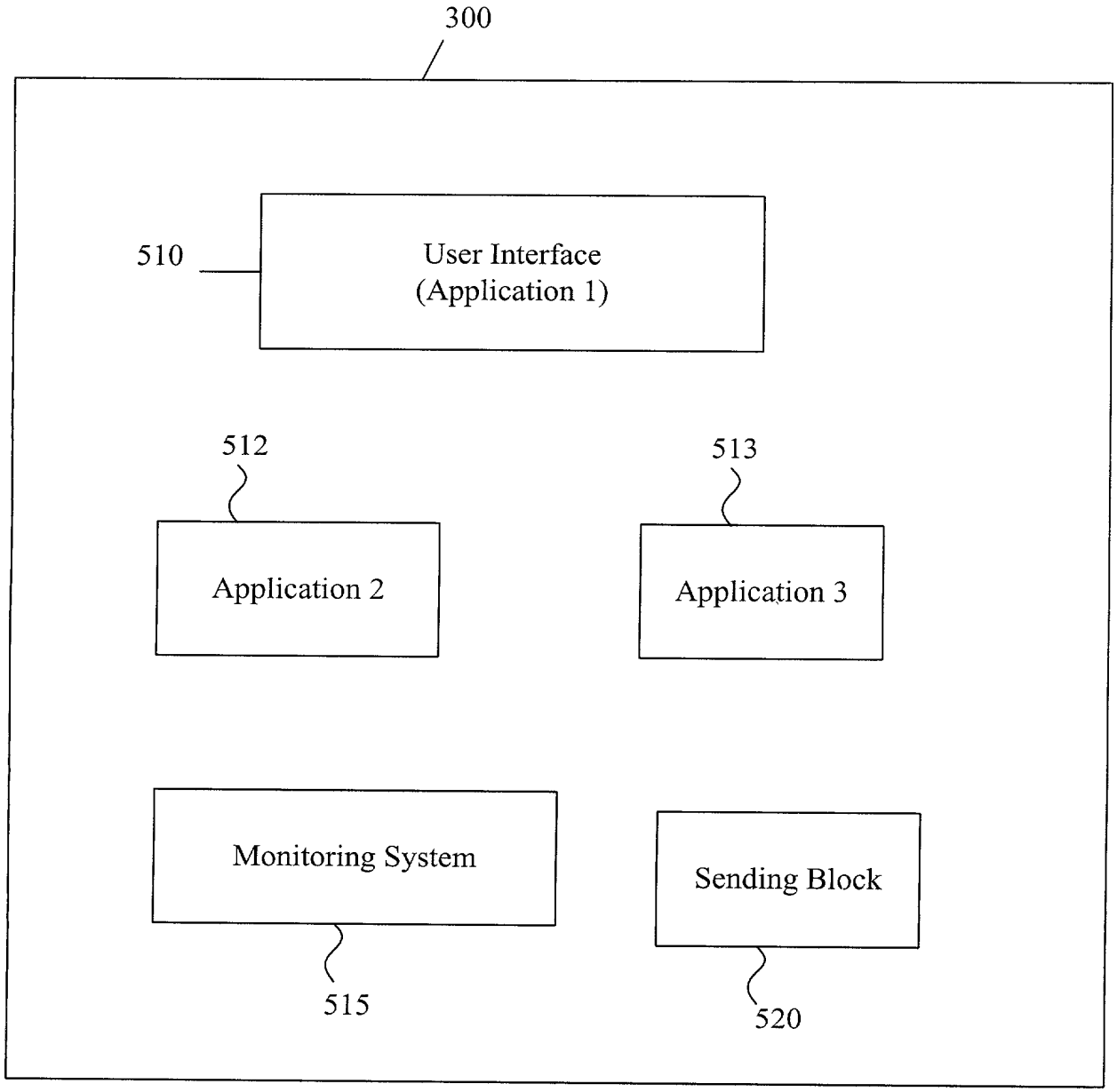

300
510
User Interface
(Application 1)
512
Application 2
513
Application 3
Monitoring System
515
Sending Block
520

FIG. 9

FIG. 10

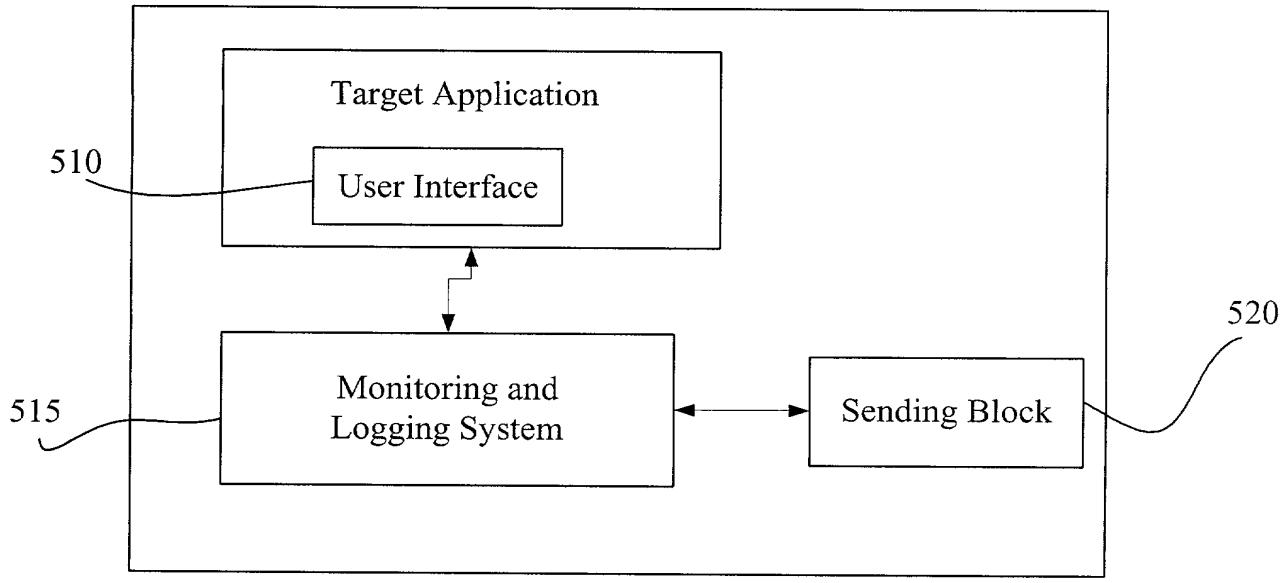
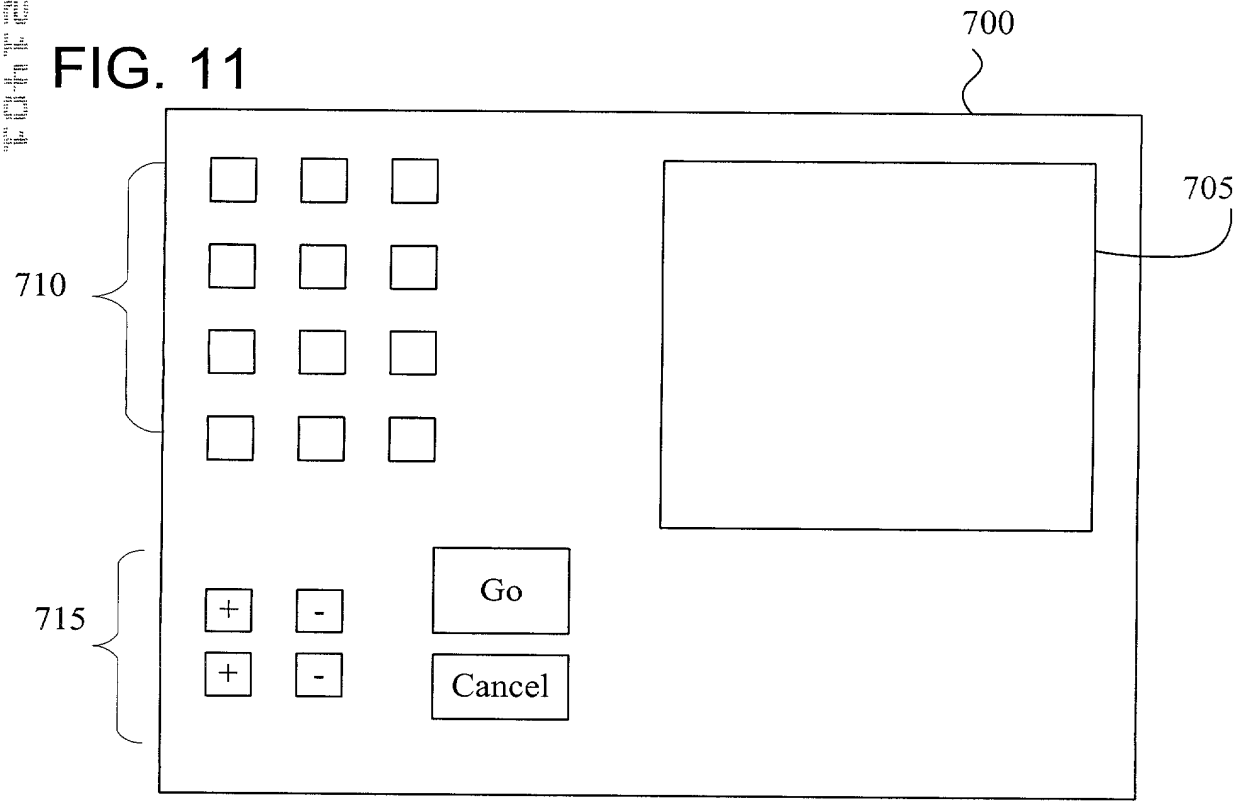

Target Application
510
User Interface
520
Monitoring and
Logging System
515
Sending Block

FIG. 11


700
705
710
715
Go
Cancel

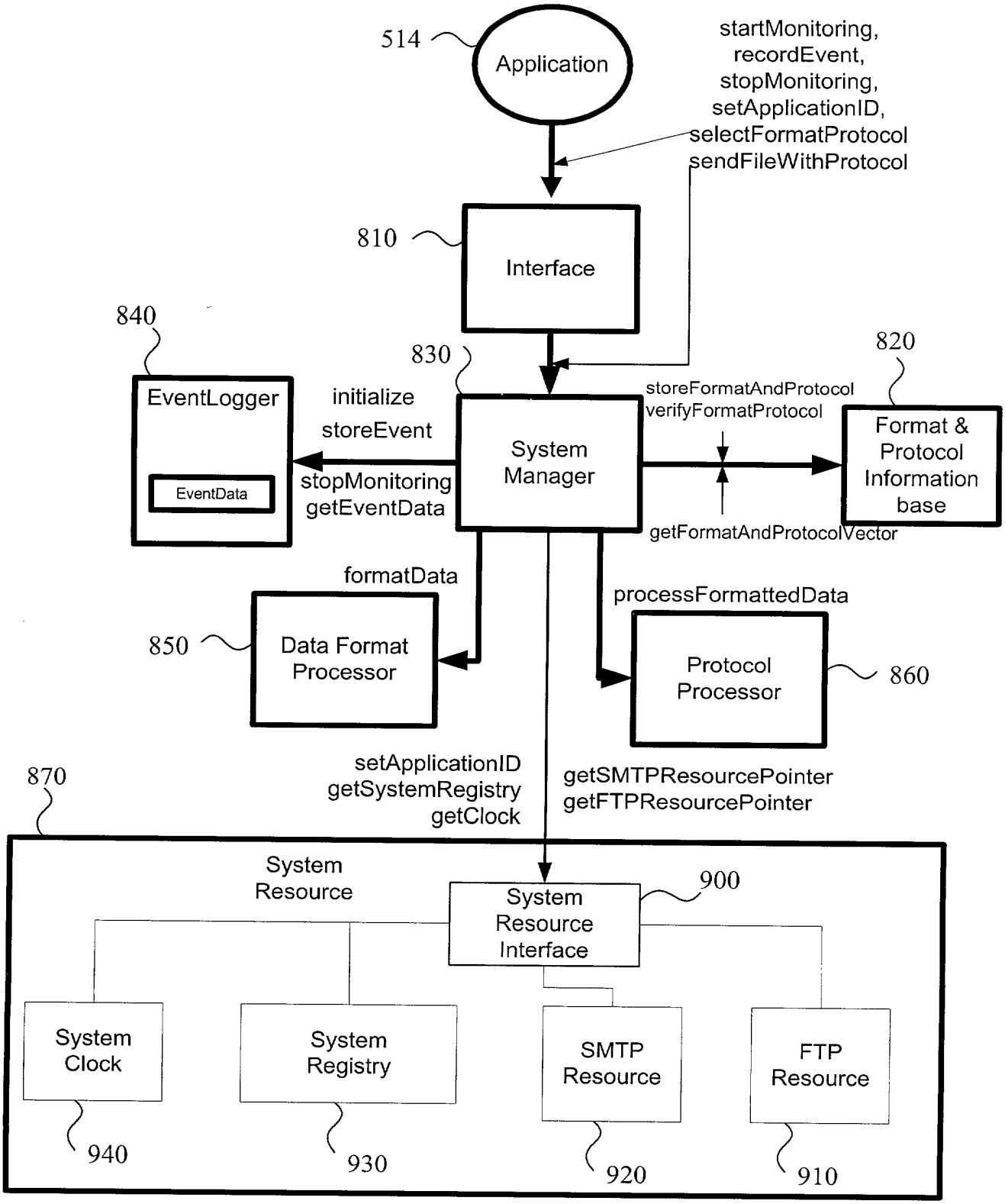

514
Application
startMonitoring,
recordEvent,
stopMonitoring,
setApplicationID,
selectFormatProtocol
sendFileWithProtocol
810
Interface
840
830
820
EventLogger
EventData
initialize
storeEvent
stopMonitoring
getEventData
System
Manager
storeFormatAndProtocol
verifyFormatProtocol
getFormatAndProtocolVector
Format &
Protocol
Information
base
formatData
processFormattedData
850
Data Format
Processor
Protocol
Processor
860
870
setApplicationID
getSystemRegistry
getClock
getSMTPResourcePointer
getFTPResourcePointer
System
Resource
900
System
Resource
Interface
System
Clock
System
Registry
SMTP
Resource
FTP
Resource
940
930
920
910

FIG. 12A

Return Value	Function Name	Description
bool	getNextSession	Returns false when there is no more session; true otherwise
string	getFileName	Returns file name for the EventData
map<string, string>	getSessionInformation	Returns the map. Keys are UserID, ApplicationID, CumulativeSessionNumber, StartTime, and Duration.
map<string, vector<string>>	getSessionEventData	Returns the map. Keys are EventName and EventTiming. The values of EventTiming vector are in the unit of 10th of a second converted from unsigned integer to string.

FIG. 12B

Return Value	Function Name	Description
bool	getNextLine	Returns one line of string data as an out parameter string. The function returns true if there is a line; false if no more line exists with empty string.
string	getFileNameWithSuffix	Returns file name for the data with suffix if applicable
enum	getDataType	Returns the data type, BINARY or TEXT

FIG. 12C

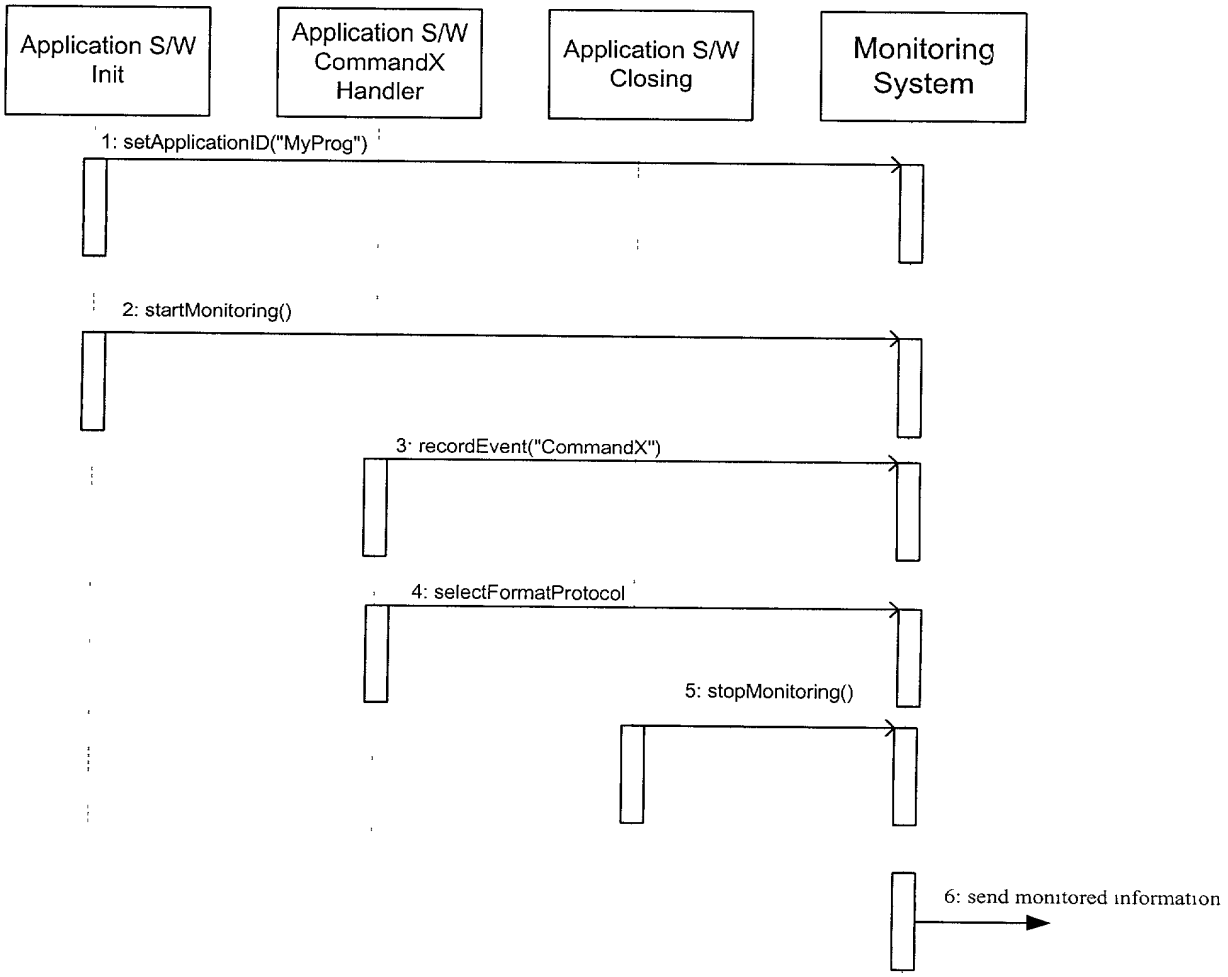

Application S/W Init
Application S/W CommandX Handler
Application S/W Closing
Monitoring System
1: setApplicationID("MyProg")
2: startMonitoring()
3: recordEvent("CommandX")
4: selectFormatProtocol
5: stopMonitoring()
6: send monitored information

FIG. 13A

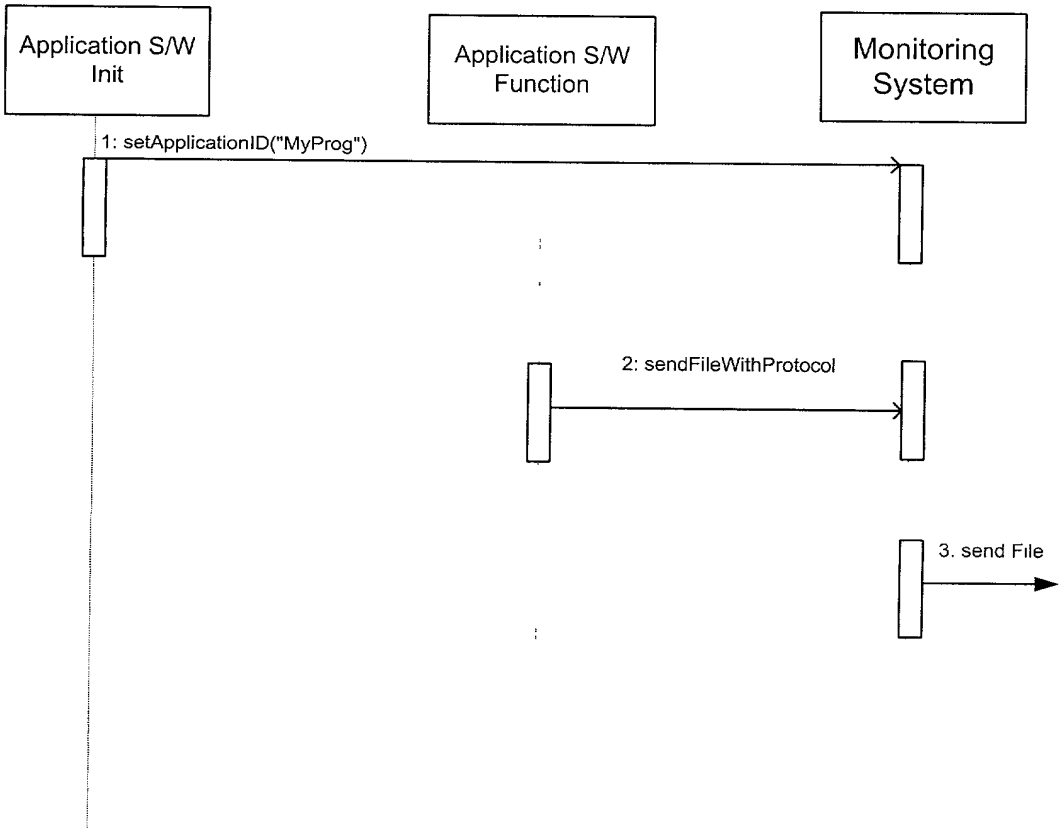

Application S/W Init
Application S/W Function
Monitoring System
1: setApplicationID("MyProg")
2: sendFileWithProtocol
3. send File

FIG. 13B

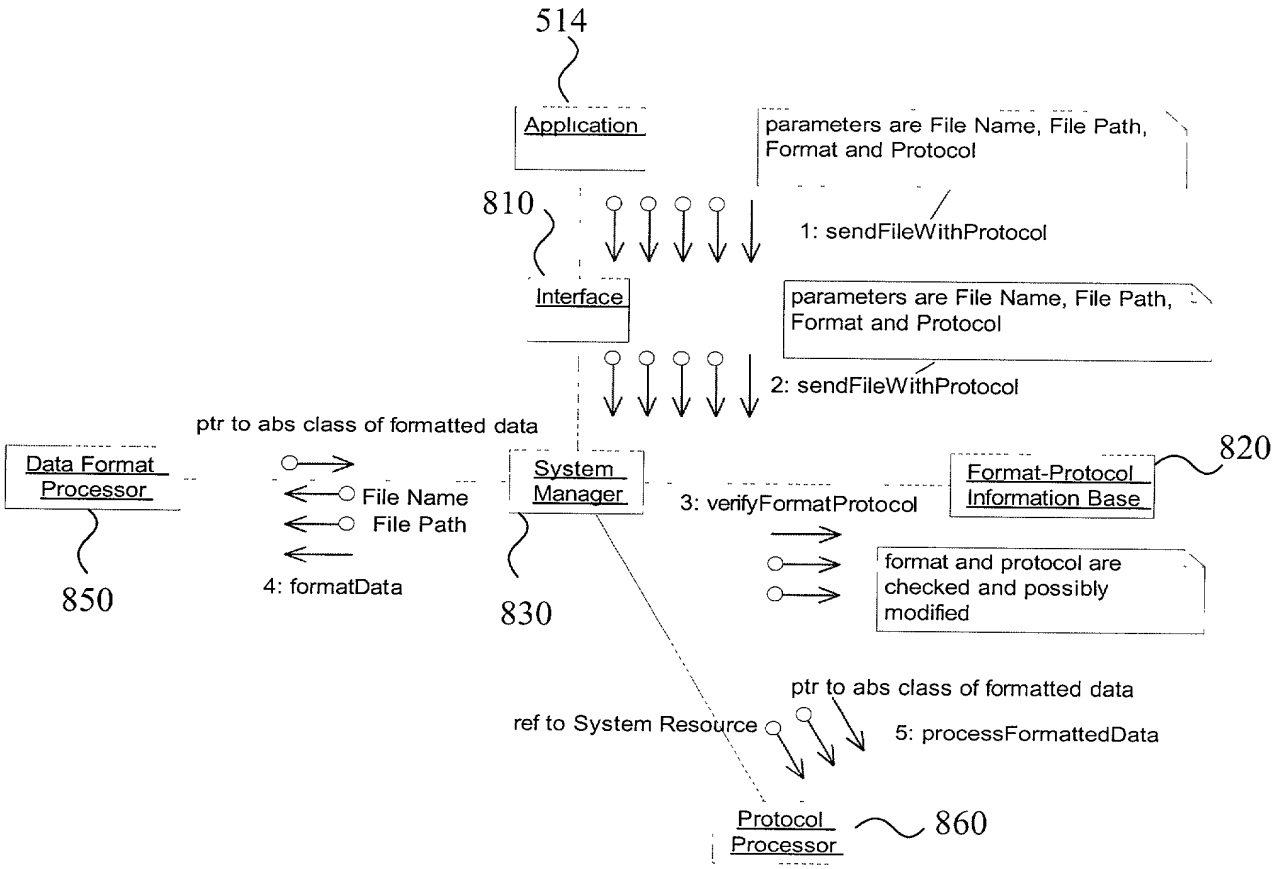

514
Application
parameters are File Name, File Path, Format and Protocol
810
1: sendFileWithProtocol
Interface
parameters are File Name, File Path, Format and Protocol
2: sendFileWithProtocol
ptr to abs class of formatted data
Data Format Processor
File Name
File Path
4: formatData
System Manager
3: verifyFormatProtocol
Format-Protocol Information Base
820
format and protocol are checked and possibly modified
850
830
ptr to abs class of formatted data
ref to System Resource
5: processFormattedData
Protocol Processor
860

FIG. 14

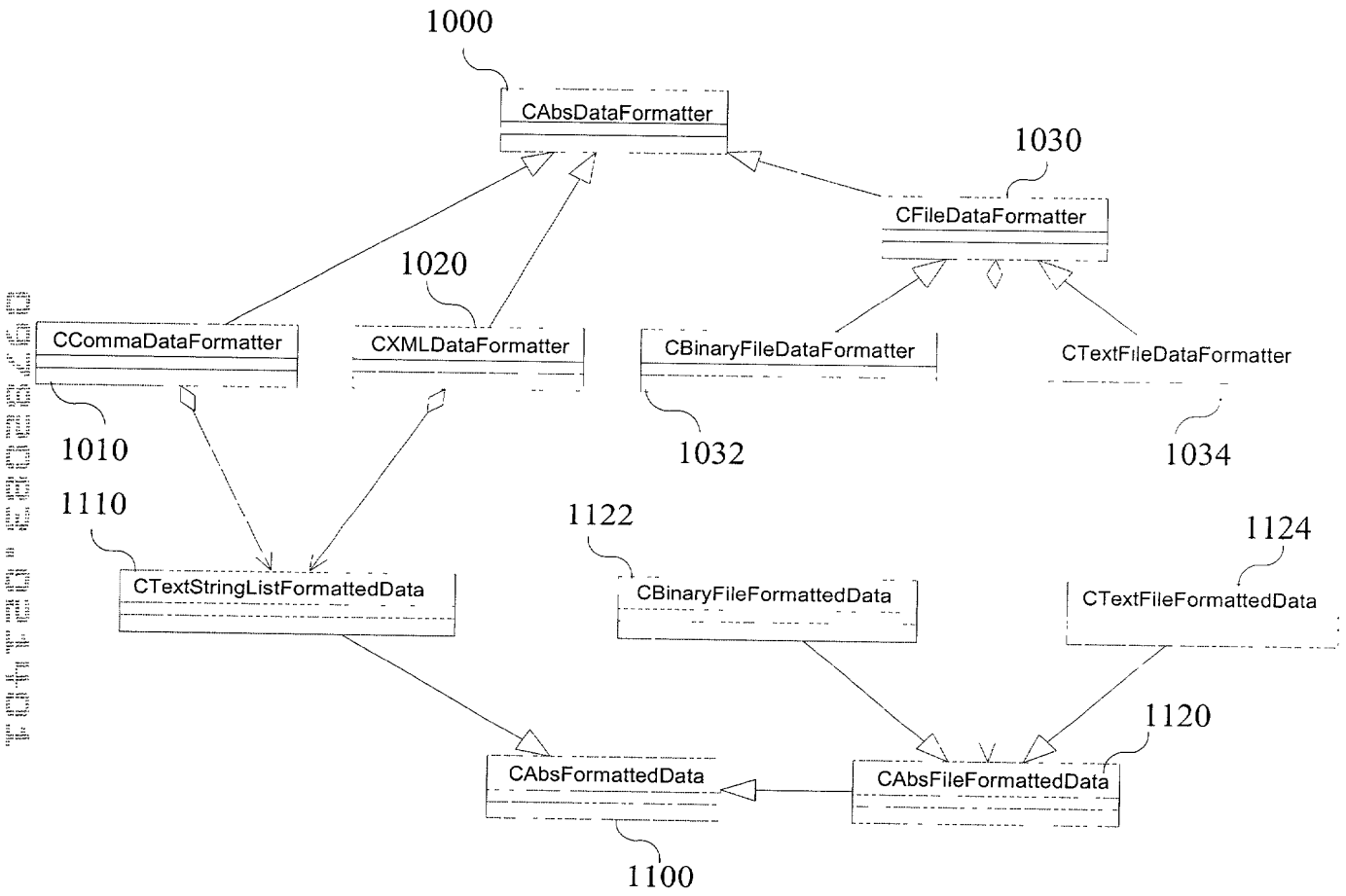

1000
CAbsDataFormatter
1030
CFileDataFormatter
1020
CCommaDataFormatter
CXMLDataFormatter
CBinaryFileDataFormatter
CTextFileDataFormatter
1010
1032
1034
1110
1122
1124
CTextStringListFormattedData
CBinaryFileFormattedData
CTextFileFormattedData
1120
CAbsFormattedData
CAbsFileFormattedData
1100

FIG. 15

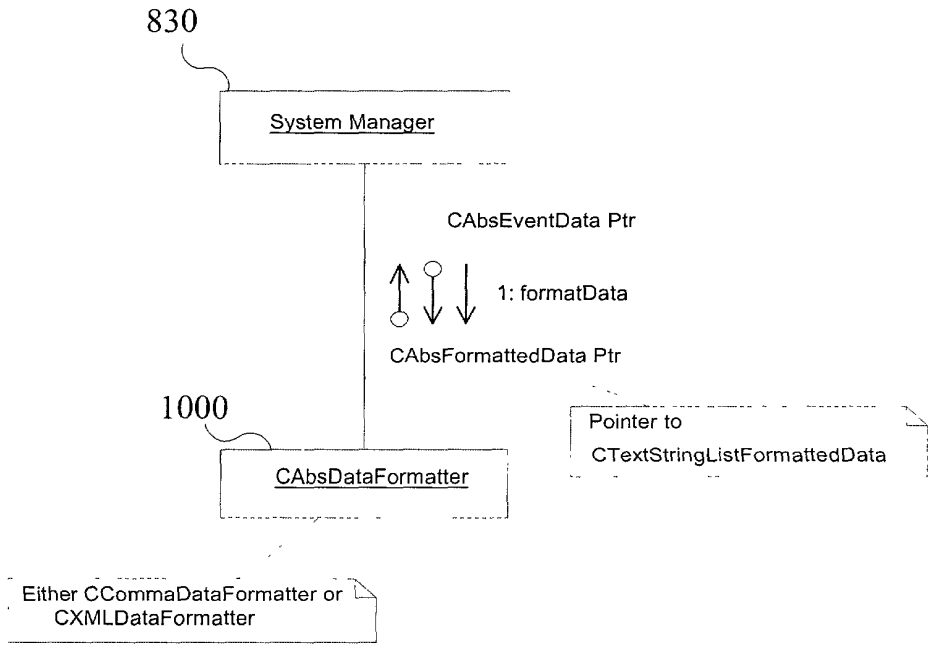

830
System Manager
CAbsEventData Ptr
1: formatData
CAbsFormattedData Ptr
1000
CAbsDataFormatter
Pointer to
CTextStringListFormattedData
Either CCommaDataFormatter or
CXMLDataFormatter

FIG. 16

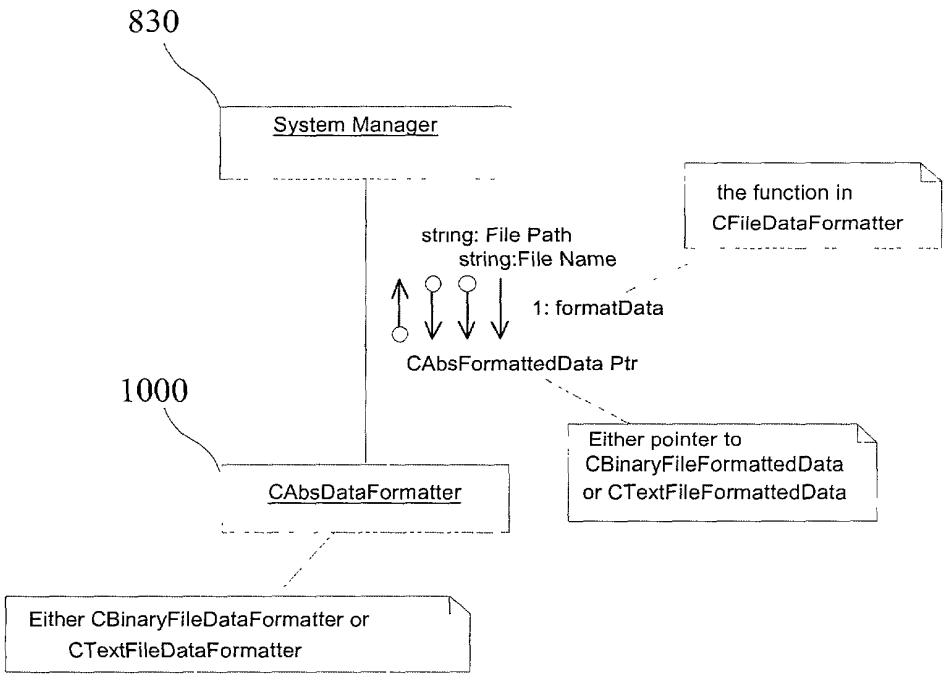

830
System Manager
the function in
CFileDataFormatter
string: File Path
string:File Name
1: formatData
CAbsFormattedData Ptr
1000
Either pointer to
CBinaryFileFormattedData
or CTextFileFormattedData
CAbsDataFormatter
Either CBinaryFileDataFormatter or
CTextFileDataFormatter

FIG. 17

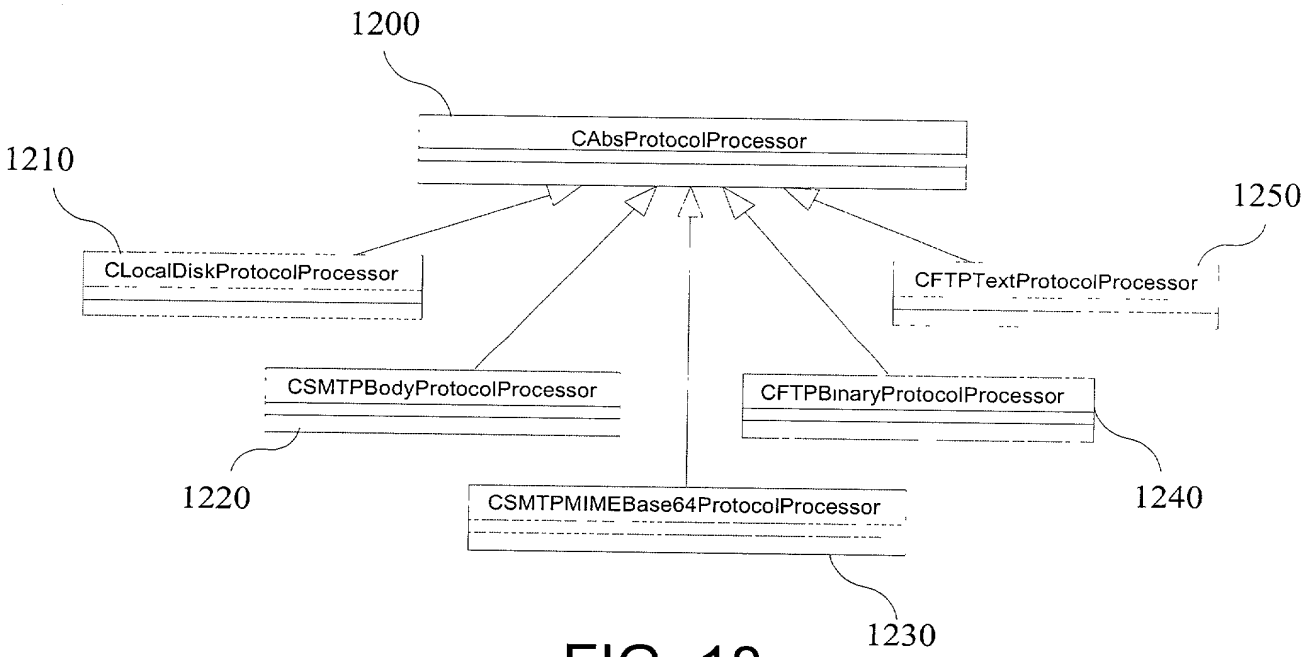

1200
CAbsProtocolProcessor
1210
CLocalDiskProtocolProcessor
1250
CFTPTextProtocolProcessor
CSMTPBodyProtocolProcessor
CFTPBinaryProtocolProcessor
1220
1240
CSMTPMIMEBase64ProtocolProcessor
1230

FIG. 18

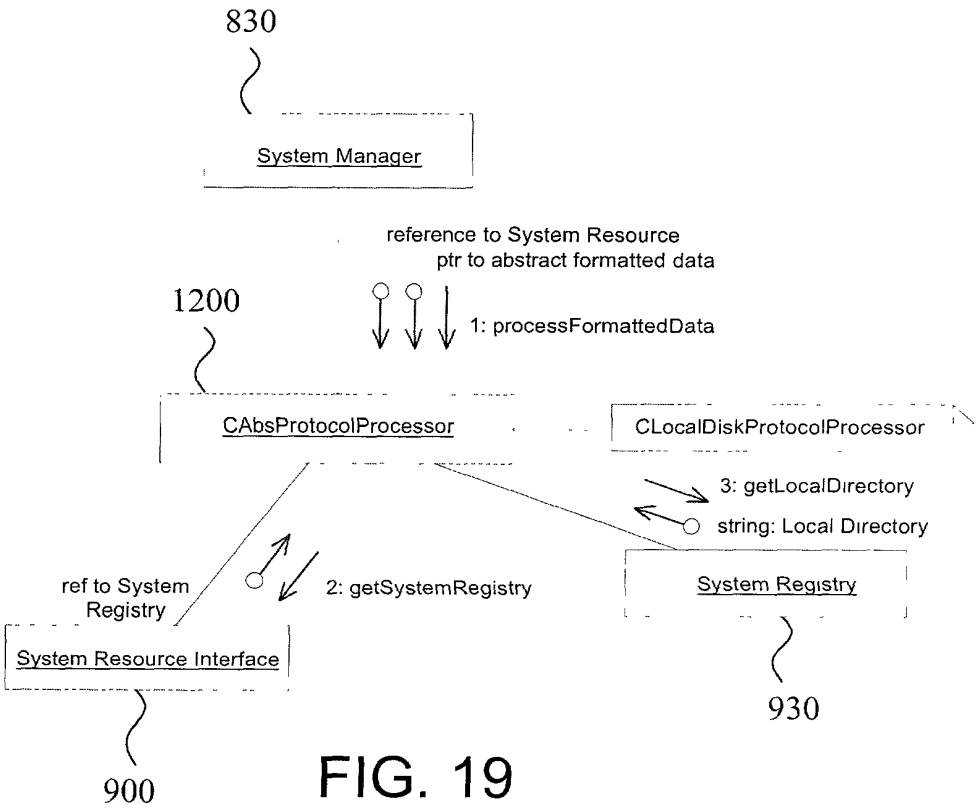

830
System Manager
reference to System Resource
ptr to abstract formatted data
1200
1: processFormattedData
CAbsProtocolProcessor
CLocalDiskProtocolProcessor
3: getLocalDirectory
string: Local Directory
ref to System Registry
2: getSystemRegistry
System Registry
System Resource Interface
930
900

FIG. 19

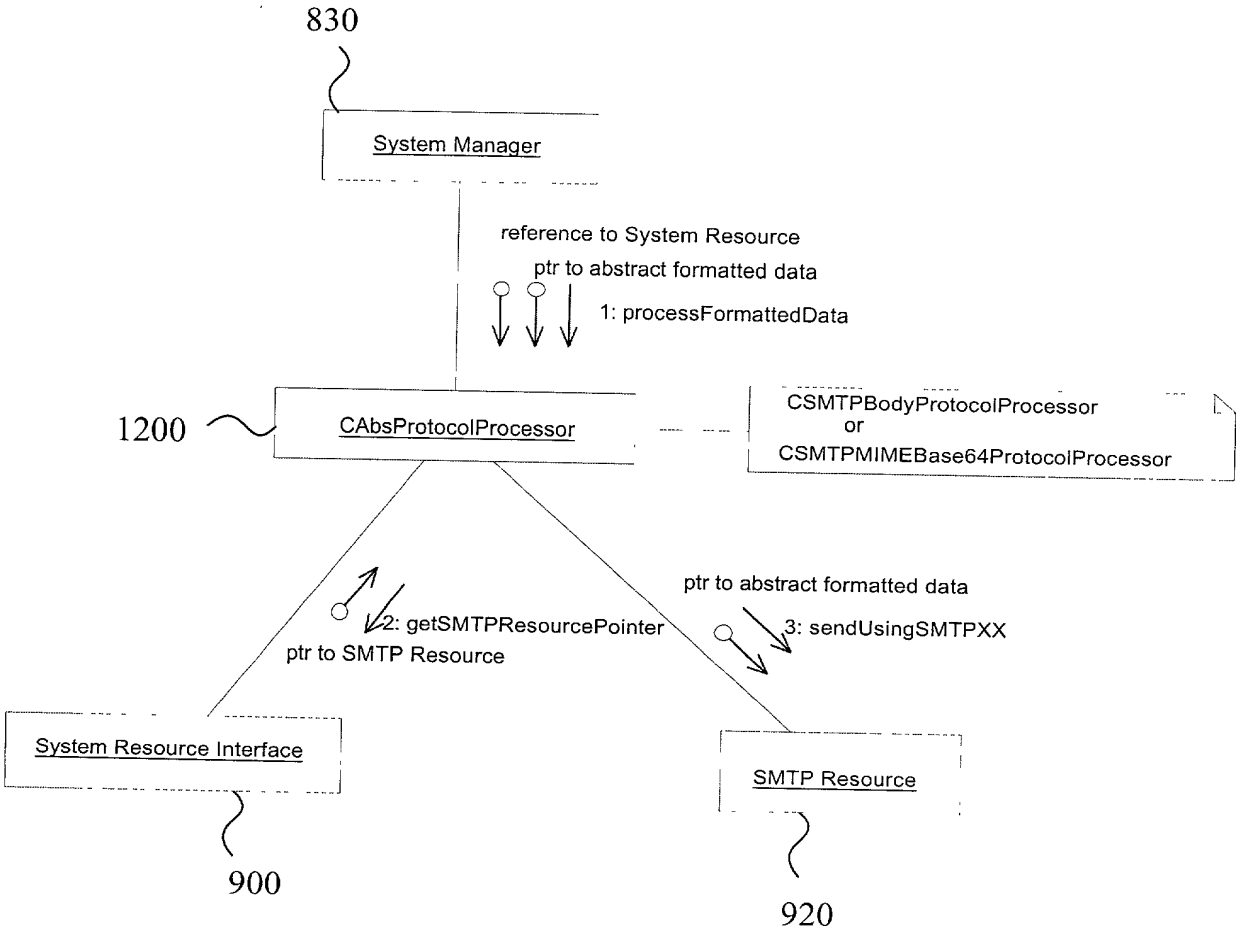

830
System Manager
reference to System Resource
ptr to abstract formatted data
1: processFormattedData
1200
CAbsProtocolProcessor
CSMTPBodyProtocolProcessor
or
CSMTPMIMEBase64ProtocolProcessor
2: getSMTPResourcePointer
ptr to SMTP Resource
ptr to abstract formatted data
3: sendUsingSMTPXX
System Resource Interface
SMTP Resource
900
920

FIG. 20

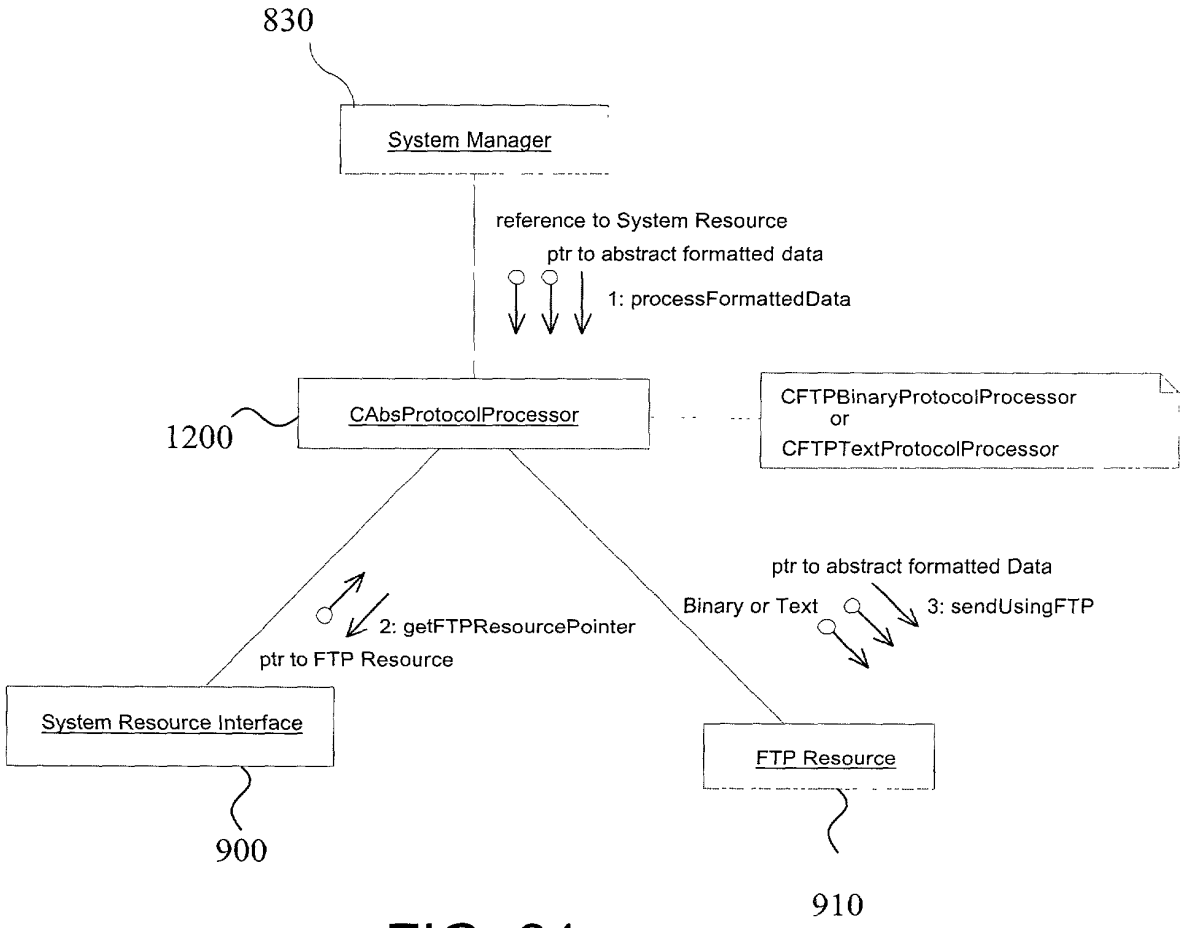

830
System Manager
reference to System Resource
ptr to abstract formatted data
1: processFormattedData
1200
CAbsProtocolProcessor
CFTPBinaryProtocolProcessor
or
CFTPTextProtocolProcessor
ptr to abstract formatted Data
Binary or Text
3: sendUsingFTP
2: getFTPResourcePointer
ptr to FTP Resource
System Resource Interface
FTP Resource
900
910

FIG. 21

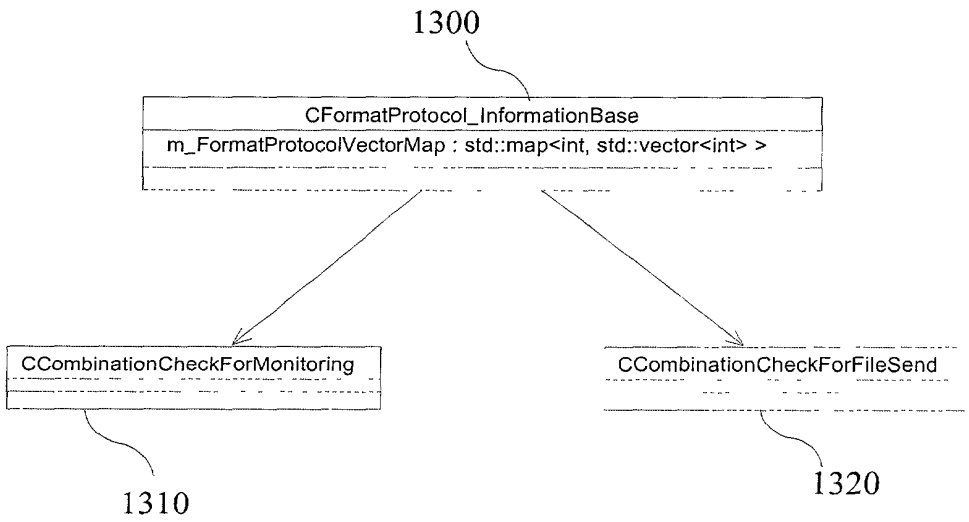

1300
CFormatProtocol_InformationBase
m_FormatProtocolVectorMap : std::map<int, std::vector<int> >
CCombinationCheckForMonitoring
CCombinationCheckForFileSend
1310
1320

FIG. 22

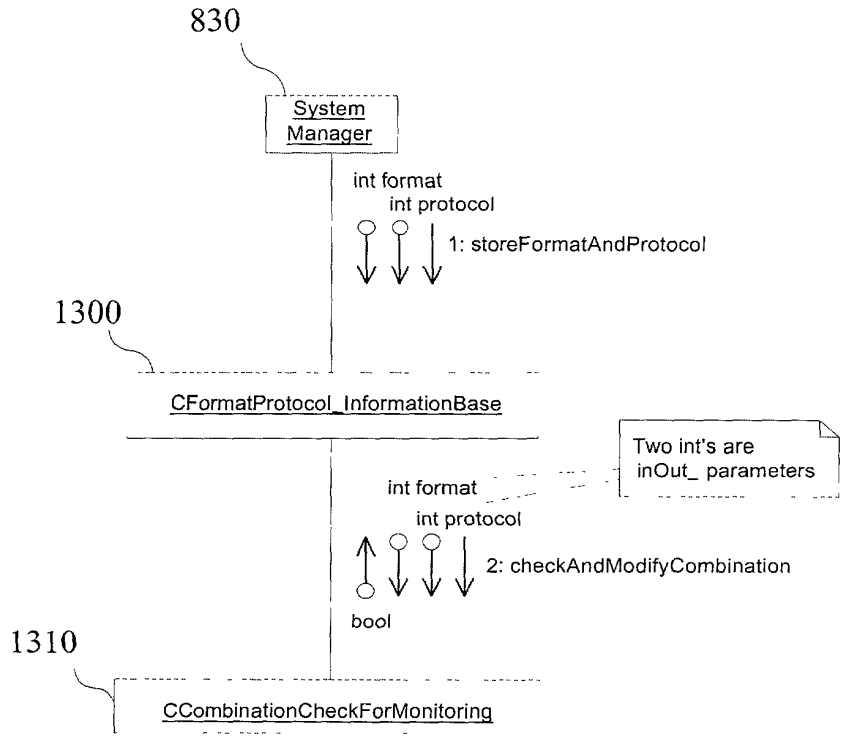

830
System Manager
int format
int protocol
1: storeFormatAndProtocol
1300
CFormatProtocol_InformationBase
Two int's are inOut_ parameters
int format
int protocol
2: checkAndModifyCombination
bool
1310
CCombinationCheckForMonitoring

FIG. 23

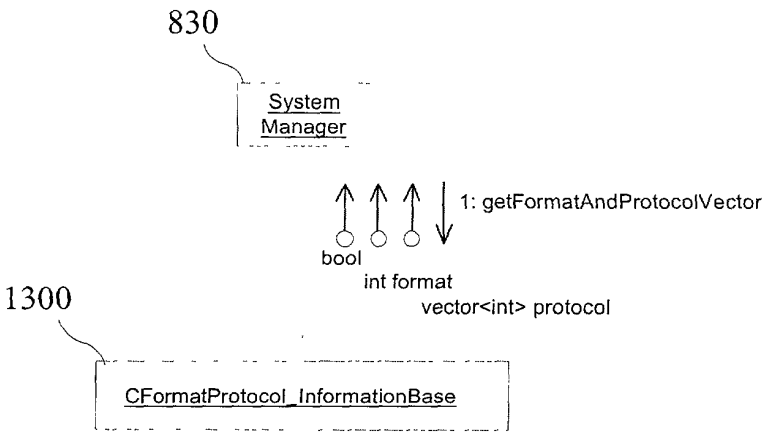

830
System Manager
1: getFormatAndProtocolVector
bool
int format
vector<int> protocol
1300
CFormatProtocol_InformationBase

FIG. 24

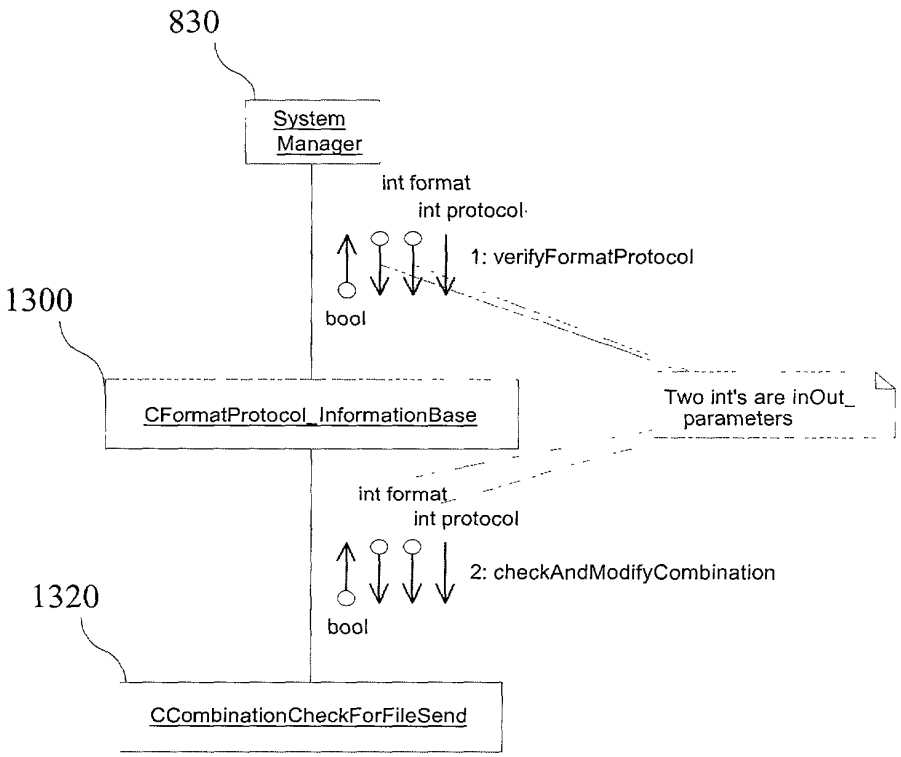

830
System Manager
int format
int protocol
1: verifyFormatProtocol
bool
1300
CFormatProtocol_InformationBase
Two int's are inOut_ parameters
int format
int protocol
2: checkAndModifyCombination
bool
1320
CCombinationCheckForFileSend

FIG. 25

Key	Value
10	{ 1 10 30 100 105 }
20	{ 1 10 30 100 105 }

FIG. 26A

1. Set return-bool to true
2. Use find function of the Map for inOut_nFormat
3. If returned iterator is end (Not found), set inOut_nFormat to the default value(10) and set return-bool to false
4. get the Set value for the key format
5. Use the find function of the Set for inOut_nProtocol
6. if returned iterator is end (Not found), set inOut_nProtocol to default (1) and set return-bool to false
7. return return-bool

FIG. 26B

Key	Value
1	<table><tr><th>Key</th><th>Value</th></tr><tr><td>1</td><td>1</td></tr><tr><td>10</td><td>10</td></tr><tr><td>30</td><td>30</td></tr><tr><td>100</td><td>100</td></tr><tr><td>105</td><td>105</td></tr></table>
5	<table><tr><th>Key</th><th>Value</th></tr><tr><td>1</td><td>1</td></tr><tr><td>10</td><td>30</td></tr><tr><td>30</td><td>30</td></tr><tr><td>100</td><td>105</td></tr><tr><td>105</td><td>105</td></tr></table>

FIG. 27A

```
1.  Set return-bool = true
2.  Use find function of the Map for inOut_nFormat
3.  If returned iterator is end, set inOut_nFormat to
    the default value (5) and set return-bool to false
4.  Get the Map corresponding to the key format
5.  Use the find function of the Map for inOut_nProtocol
6.  if returned iterator is end{
        set inOut_nProtocol to default (105)
        and set return-bool to false
        }
    else {
        return-bool = (inOut_nProtocol EQ
                    (Value field corresponding to
                    inOut_nProtocol))
                   logical-AND return-bool.
        set inOut_nProtocol =
        (Value field corresponding to inOut_nProtocol).
        }
7.  return return-bool
```

FIG. 27B